CLAIMS

I claim:

1. A method for maximizing CPU performance in a multiprocessor, comprising:
   (a) allowing local memory operations to execute in an arbitrary order; and
   (b) providing execution constraints for shared memory operations.

2. The method of claim 1, further comprising assigning first and second registers of a CPU for storing associated first and second instruction addresses.

3. The method of claim 2, further comprising providing a third instruction referencing said registers.

4. The method of claim 3, wherein said third instruction specifies ordering between said first and second instructions.

5. The method of claim 4, wherein said third instruction indicates said first instruction's execution attaining a first specified state of execution prior to said second instruction's execution attaining a second specified state of execution.

6. The method of claim 5, wherein said first and said second specified states of execution are selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/O access, completing an I/O access, and completing instruction execution.

7. The method of claim 1, further comprising assigning a sequence number to an associated instruction for maintaining instruction ordering.

8. The method of claim 7, further comprising statically encoding said sequence number within said instruction.

9. The method of claim 7, further comprising dynamically encoding said sequence number within said instruction.

10. The method of claim 1, further comprising placing a range of instructions into a hierarchical ordering system.

11. The method of claim 10, further comprising implementing a special instruction for maintaining a hierarchical execution of said instruction.

12. A processor for use in a multiprocessor computer system, comprising:
a first instruction for allowing local memory operations to occur in an arbitrary order; and
a second instruction for providing shared memory operation constraints.

13. The processor of claim 12, further comprising a first register to store a first instruction address and a second register to store a second instruction address.

14. The processor of claim 13, further comprising a third instruction to manage order of execution of said first and second instructions.

15. The processor of claim 14, wherein execution of said second instruction is responsive to said first instruction reaching a specified state of execution.

16. The processor of claim 15, wherein said specified state of execution is selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/O access, completing an I/O access, and completing instruction execution.

17. The processor of claim 12, wherein said first and second instructions are assigned a sequence number to specify an order of instruction execution.

18. The processor of claim 17, wherein said sequence number is statically encoded within said instruction.

19. The processor of claim 17, wherein said sequence number is dynamically encoded within said instruction.

20. The processor of claim 12, further comprising a manager to place a range of instructions in a hierarchical order.

21. The processor of claim 19, further comprising a special instruction to maintain execution of said instruction in said hierarchical order.

22. A processor for use in a multiprocessor computer system, comprising:
a first instruction for allowing local memory operations to occur in an arbitrary order;
a second instruction for providing shared memory operation constraints;
a third instruction for managing order of execution of said first and second instructions;
wherein execution of said second instruction is responsive to said first instruction reaching a specified state of execution and said specified state of execution is selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/O access, completing an I/O access, and completing instruction execution.

23. The processor of claim 22, a first register to store a first instruction address and a second register to store a second instruction address.

24. The processor of claim 22, wherein said first and second instructions are assigned a sequence number to specify an order of instruction execution.

25. The processor of claim 22, further comprising a special instruction to maintain execution of said instructions in said hierarchical order.